IN THE SPECIFICATION:

On page 1, please replace the first paragraph with the following amended paragraph:

--This application is a continuation-in-part of U.S. Patent Application Serial No. 09/518,492 titled "System and Method for Programmatically Creating a Graphical Program", filed March 3, 2000, whose inventors were Ram Kudukoli, Robert Dye, Melanie Jensen, and Yumiko Kawachi, which claims benefit of priority of U.S. provisional Patent Application Serial No. ~~60/149,943~~ 60/149,942 titled "System and Method for Programmatically Creating a Graphical Program" and filed August 19, 1999.--

a1

On page 1, please replace the paragraph beginning on line 19 with the following amended paragraph:

--Traditionally, high level text-based programming languages have been used by programmers in writing application programs. Many different high level programming languages exist, including BASIC, C, ~~Java~~ JAVA, FORTRAN, Pascal, COBOL, ADA, APL, etc. Programs written in these high level languages are translated to the machine language level by translators known as compilers or interpreters. The high level programming languages in this level, as well as the assembly language level, are referred to herein as text-based programming environments.--

a2

On page 3, please replace the paragraph beginning on line 6 with the following amended paragraph:

--U.S. Patent Nos. 4,901,221; 4,914,568; 5,291,587; 5,301,301; and 5,301,336; among others, to Kodosky et al disclose a graphical system and method for modeling a process, i.e., a graphical programming environment which enables a user to easily and intuitively model a process. The graphical programming environment disclosed in Kodosky et al can be considered a higher and more intuitive way in which to interact with a computer.

a3

a3 A graphically based programming environment can be represented at a level above text-based high level programming languages such as C, Basic, ~~Java~~ JAVA, etc.--

On page 27, please replace the paragraph beginning on line 14 with the following amended paragraph:

--Figure 4 is a flowchart diagram illustrating one embodiment of a method for programmatically generating a graphical program. In step 200, a graphical program generation (GPG) program may be created, wherein the GPG program is operable to programmatically generate a plurality of graphical programs, based on received information. As described below, the GPG program may be associated with any of various purposes or applications. Also, as discussed above, the GPG program may be

a4 implemented in various ways, e.g., using graphical and/or text-based programming environments. For example, the GPG program may be a text-based program, such as a program written using C, C++, ~~Java~~ JAVA, Basic, Visual Basic, FORTRAN, Pascal, or another text-based programming language. Also, the GPG program may itself be a graphical program. For example, the GPG program may be a graphical program interactively created in response to user input.--

On page 39, please replace the paragraph beginning on line 11 with the following amended paragraph:

--As noted above, the client program 502 may be any of various types of programs. For example, the client program 502 may be a graphical program. The client program 502 may also be a text-based program such as a C++ program, a Visual Basic

a5 program, a ~~Java~~ JAVA program, etc., or any combination of these or other languages. The client program 502 may execute independently or may execute within an execution subsystem of an application development environment.—

On page 59, please replace the paragraph beginning on line 6 with the following amended paragraph:

--In the preferred embodiment, various types of software components are provided which enable programs of various languages executing on various systems to programmatically create/edit graphical programs. As is well known in the art, modern component-based software architecture and object oriented design techniques make it relatively straightforward to encapsulate portions of code and provide various interfaces to the code. For example, a service to create/edit a graphical program may be exposed as an ActiveX component, a CORBA component, a ~~Java~~ JAVA component, etc. In one embodiment, the implementation of the nodes described above and the implementation of components for use in text-based programs may be based on a common code base.--

a6

On page 73, please replace the abstract with the following new abstract:

--A system and method for programmatically generating a graphical program or a portion of a graphical program in response to receiving program information is disclosed. During execution of a graphical program generation (GPG) program, the GPG program receives program information specifying functionality of the graphical program to be generated. In one embodiment the program information does not specify specific nodes in the graphical program or connections among the nodes. In response to the program information, the GPG program programmatically generates a graphical program (or graphical program portion) that implements the specified functionality.--

a7